



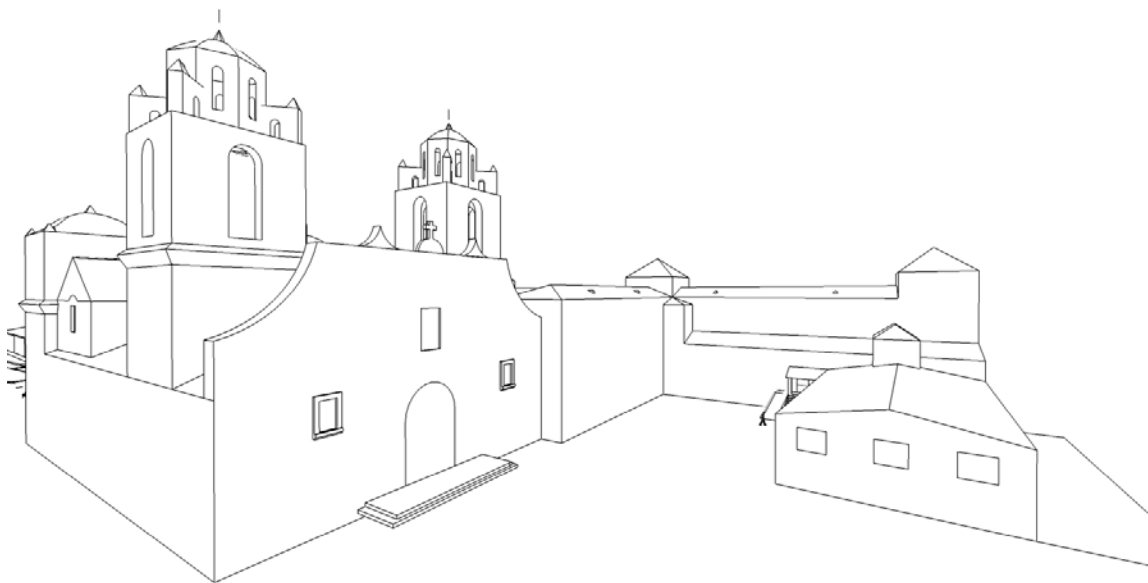
Guía Docente 2014/2015

Programación Orientada a Objetos

Object Oriented Programming

Grado en Ingeniería Informática

Modalidad Presencial



Índice

Programación Orientada a Objetos.....	3
Breve descripción de la asignatura	3
Requisitos Previos	3
Objetivos	3
Competencias y resultados de aprendizaje	4
Metodología	5
Temario.....	6
Relación con otras materias.....	8
Sistema de evaluación	8
Bibliografía y fuentes de referencia.....	9
Web relacionadas	9
Recomendaciones para el estudio.....	10
Material necesario	10
Tutorías	12

Programación Orientada a Objetos

Programación Orientada a Objetos

Módulo: **Común de la rama de informática.**

Materia: **Programación.**

Carácter: **Obligatoria.**

Nº de créditos: **4.5 ECTS.**

Unidad Temporal: **2º Curso – 1er Semestre**

Profesor/a de la asignatura: **María Teresa García Valverde**

Email: **mtvalverde@ucam.edu**

Horario de atención a los alumnos/as: **Martes de 12:00 a 13:00** (si el alumno no es capaz de atender en este horario, puede **solicitar una tutoría** vía la herramienta de mensajería privada del campus virtual)

Profesor coordinador de curso: **Andrés Bueno Crespo**

Profesor coordinador de módulo: **Andrés Muñoz Ortega**

Breve descripción de la asignatura

En la esta asignatura abordaremos una estrategia de programación conocida como programación orientada a objetos, que permite una mayor flexibilidad a la hora del diseño y la implementación de nuevas aplicaciones. Entre los conceptos que se cubrirán en la asignatura están los conceptos de herencia, polimorfismo, ligadura dinámica, genericidad, etc. Factores de calidad del software. Patrones de diseño. Extensibilidad. Reutilización. Rendimiento. Portabilidad. Sencillez.

Brief Description

This subject covers the fundamentals of object-oriented programming, a programming strategy that allows us an easy way of designing and implementing new applications. Among the main concepts behind this technique, we will study inheritance, polymorphism, dynamic binding, genericity, etc. Software quality factors. Extendibility. Reusability. Portability. Ease of use.

Requisitos Previos

Son necesarios conocimientos básicos de programación adquiridos en las asignaturas de Fundamentos de programación I y Fundamentos de programación II.

Objetivos

Los objetivos específicos de la asignatura son:

1. Conocer el paradigma de programación orientada a objetos.
2. Diferenciar entre programación estructurada y programación orientada a objetos.
3. Enumerar y explicar los distintos conceptos de la programación orientada a objetos.

Programación Orientada a Objetos

4. Aplicar estructuras coherentes basadas en programación orientada a objetos para la resolución de problemas reales.
5. Entender la reutilización y la herencia.
6. Aplicar soluciones genéricas a problemas concretos mediante el uso de patrones de diseño.
7. Conocer el lenguaje JAVA.
8. Identificar adecuadamente los distintos conceptos que se proponen en el temario.

Competencias y resultados de aprendizaje

Competencias transversales

- T1 - Capacidad de análisis y síntesis.
- T3 - Capacidad de gestión de la información.
- T4 - Resolución de problemas.
- T5 - Toma de decisiones.
- T6 - Trabajo en equipo.
- T14 - Aprendizaje autónomo.
- T15 - Adaptación a nuevas situaciones.
- T16 - Creatividad e innovación.
- T21 - Capacidad de reflexión.

Competencias específicas

- C6 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

Resultados de aprendizaje

- RA 2.3.1. Comprender el paradigma de programación orientada a objetos.
- RA 2.3.2. Distinguir entre el paradigma de programación estructurada y el de programación orientada a objetos.

Programación Orientada a Objetos

RA 2.3.3. Explicar los conceptos de herencia, polimorfismo y ligadura dinámica desde un punto de vista teórico y su aplicación práctica.

RA 2.3.4. Diseñar estructuras coherentes basadas en programación orientada a objetos para la resolución de problemas reales.

RA 2.3.5. Aplicar soluciones genéricas a problemas concretos mediante el uso de patrones de diseño.

RA 2.3.6. Desarrollar aplicaciones informáticas utilizando un lenguaje de programación orientado a objetos.

Metodología

Metodología	Horas	Horas de trabajo presencial	Horas de trabajo no presencial
Clases en el aula	13.5	45 horas (40 %)	
Evaluación	2.25		
Prácticas	20,25		
Tutorías	9		
Estudio personal	20.25		67.5 horas (60 %)
Lecturas recomendadas y búsqueda de información	3.375		
Realización de ejercicios, presentaciones, trabajos y casos prácticos	40.5		
Actividades de aprendizaje virtual	3.375		
TOTAL	112.5	45	67.5

Programación Orientada a Objetos

En el plan de trabajo de la asignatura está reflejada la distribución en el tiempo de cada metodología por tema/tarea de acuerdo al calendario académico.

Temario

Programa de la enseñanza teórica

Tema 1. Introducción a la orientación a objetos.

1. Calidad del software

Tema 2. El camino hacia la orientación a objetos.

1. Introducción
2. Modularidad
3. Reutilización
4. Tipos abstractos de datos

Tema 3 Conceptos de orientación a objetos

1. Clase
2. Objeto
3. Método
4. Mensaje
5. Semántica referencia

Tema 4. Herencia, Polimorfismo, Genericidad, Interfaces, Colecciones en Java

1. Herencia y creación
2. Herencia y Ocultamiento de la Información
3. Redefinición de características
4. Polimorfismo
5. Herencia y Sistema de tipos
6. Ligadura dinámica

Programación Orientada a Objetos

7. Clase Object
8. Genericidad y Herencia.
9. Genericidad restringida
10. Clases abstractas
11. Interfaces

Tema 5 Patrones de diseño

1. Método Plantilla
2. Patrón Estrategia
3. Patrón Composite

Tema 6 Corrección y Robustez

1. Asertos y pruebas unitarias
2. Excepciones

Programa de la enseñanza práctica

El objetivo de las prácticas es desarrollar los conceptos teóricos del paradigma de programación orientada a objetos que han sido impartidos. Se aprenderá a reconocer y utilizar las buenas prácticas en la programación orientada a objetos. Paralelamente el alumno conocerá el lenguaje Java.

Las prácticas serán de dos tipos: trabajos en clase y entregables/obligatorias.

Los trabajos en clase consistirán en la realización de ejercicios propuestos por la profesora asociados a cada tema visto en las horas de clase (ver plan de trabajo para la distribución y tiempos de prácticas/contenidos). Cuando se haya terminado de impartir los contenidos correspondientes, se destinará parte de la clase para la realización de los ejercicios. Entonces, se dejará unos días para la finalización de los ejercicios de forma autónoma (si fuese necesario) y se subirán las soluciones para que el alumno pueda comprobar la corrección de sus ejercicios. Así, los trabajos de clase se organizarán en 4 boletines:

- Boletín 0: Introducción a Java y Eclipse. Actividades de repaso.
- Boletín 1: Aplicación de los conceptos básicos de OO.
- Boletín 2: Aplicación de los conceptos avanzados de la OO: interfaces, colecciones, genericidad, etc.
- Boletín 3: Interfaces gráficas con Java Swing.

Programación Orientada a Objetos

Las prácticas entregables consistirán en un único caso práctico y real propuesto por la profesora. Dicho supuesto se desarrollará de forma incremental en tres entregas a lo largo de la asignatura (consultar plan de trabajo para ver las fechas):

- Práctica 1. Análisis y diseño con UML.
- Práctica 2. Implementación del modelo de la aplicación en Java.
- Práctica 3. Swing Java. Patrón Modelo Vista Controlador (MVC). Desarrollo de una interfaz gráfica en Java.

Tras cada entrega, la profesora notificará los errores y mejoras a introducir para la siguiente entrega.

El enunciado con el caso práctico junto con instrucciones para la correcta realización y entrega de las prácticas será proporcionado y notificado mediante el campus virtual y las tareas asociadas a cada entrega.

Relación con otras materias

- Fundamentos de Programación I y II.
- Algoritmia.
- Programación Paralela.
- Programación de aplicaciones distribuidas.
- Ingeniería de requisitos.
- Modelado de software.
- Calidad del software.

Sistema de evaluación

- **Primera prueba parcial:** 30% del total de la nota.

Se evaluará mediante un examen presencial consistente en una parte teórica (50%) y una parte práctica (50%).

- **Prueba final:** 30% del total de la nota.

Se evaluará mediante un examen presencial consistente en una parte teórica (50%) y una parte práctica (50%).

- **Evaluación de prácticas y problemas:** 40% del total de la nota.

Programación Orientada a Objetos

- Práctica 1: 20% de la nota de prácticas y problemas.
- Práctica 2: 40% de la nota de prácticas y problemas.
- Práctica 3: 40% de la nota de prácticas y problemas.

La profesora se reserva el derecho de mantener una entrevista personal con el alumno al finalizar la última práctica para comprobar la autoridad de las prácticas entregadas por el alumno. Una inadecuada defensa de las prácticas supondrá el suspenso de todas las prácticas.

Bibliografía y fuentes de referencia

Bibliografía básica

- Thinking in Java. Eckel, Bruce. 2006.
- Construcción de software orientado a objetos. Meyer, Bertrand. 2ª edición. Madrid: Prentice-Hall. 1999.

Bibliografía complementaria

- Introducción a la Programación orientada a objetos con JAVA. Thomas Wu, C. 1ª edición. Madrid: McGraw Hill. 2001.
- Problemas resueltos de programación en lenguaje Java. Pérez Menor, José Mª et al. 2002.
- Cómo programar en Java. Paul Deitel, Harvey Deitel. 2012.
- Programación orientada a objetos con Java. Francisco Durán, Francisco Gutiérrez, Ernesto Pimentel. 2007.
- Java SE 6. Teo, F. Javier. 2007.
- Eclipse in action : a guide for Java developers. David Gallardo. Ed Burnette, Robert McGovern. 2003.
- Java design patterns: a tutorial. James W. Cooper. 2000.
- Patrones de Diseño. Gamma, E.; Helm, R.; Jonson, R. Vlissides, J. 1ª edición. Madrid: Addison-Wesley Iberoamericana. 2002.
- Java 2: manual de usuario y tutorial. Froufe Quintas, Agustín. 1999.

Web relacionadas

Página oficial de Java: <http://www.oracle.com/technetwork/java/index.html>

Foro de desarrollo web con Java: (<http://www.javahispano.org/>)

Programación Orientada a Objetos

Documentación oficial de Oracle <http://docs.oracle.com/javase/tutorial>

Página oficial de Eclipse: <https://www.eclipse.org/>

Página de recursos UML: <http://www.uml.org/>

Recomendaciones para el estudio

Para realizar un correcto seguimiento de la asignatura el alumno debe asistir a todas las sesiones teóricas y prácticas. Además, debe comprobar, mediante la realización de ejercicios prácticos en el ordenador, que es capaz de poner en práctica los conceptos estudiados y resolver los problemas propuestos.

El alumno deberá repasar y tener claros los conceptos previos de programación a esta asignatura. Para ayudar al alumno en esta tarea se le proporcionan ejercicios resueltos de refuerzo y repaso de los contenidos previos.

Dado el carácter incremental y continuo de la asignatura, es muy importante que el alumno realice un seguimiento semanal de la asignatura para realizar un aprendizaje continuo y sumativo. Se recomienda también, dado que es una asignatura eminentemente práctica, que se compaginen los contenidos teóricos y prácticos, tal y como se distribuye en el plan de trabajo establecido para la asignatura.

Material necesario

Material didáctico

Para esta asignatura se utilizarán las aulas preparadas con ordenadores y con los programas necesarios para impartir el temario.

El alumno dispondrá mediante el campus virtual (zona recursos) del material didáctico necesario para el correcto seguimiento de la asignatura. El material será organizado en temas y tareas.

Dicho material consistirá en:

- Apuntes sobre los temas tratados.
- Ejercicios de repaso de programación estructurada con soluciones.
- Ejercicios guiados para la instalación y puesta en marcha de entorno.

Programación Orientada a Objetos

- Ejercicios de trabajo en clase con soluciones.

Además, el alumno dispondrá de la documentación de los cursos oficiales de Oracle Java SE. Dicha documentación estará disponible en la zona de recursos del campus virtual. También tendrá acceso a todo el material, ejercicios, foros, etc. de los cursos Java SE de Oracle Academy.

Por último, el alumno puede ampliar toda la información de cada tema mediante la lista de bibliografía básica y complementaria (toda disponible en la biblioteca de la UCAM) y mediante las webs relacionadas con la asignatura.

Software/Hardware

El software a utilizar será el JDK (Java Development Kit) de Java y el entorno de desarrollo Eclipse. Ambos son gratuitos y multiplataforma.

El JDK puede descargarse desde la web de Oracle:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Eclipse se puede descargar desde <http://www.eclipse.org/downloads> . Para principiantes, debe elegirse el entorno mínimo Eclipse IDE for "Java" Developers (según SSOO y procesador).

Eclipse dispone de documentación oficial online (<https://www.eclipse.org/documentation/>) y documentación en local integrada en el propio entorno. Además, existe la posibilidad de bajar desde el propio entorno los paquetes para traducir el entorno a castellano (aunque se recomienda su uso en inglés).

Los requisitos mínimos deseables para las últimas versiones del JDK y de Eclipse son 450MB de espacio en disco y 1GB de RAM.

Para la elaboración de la práctica 1 se puede opcionalmente utilizar herramientas de modelado UML como Visual Paradigm o ArgoUML. Ambas herramientas disponen de versión educativa gratuita y son multiplataforma. Pueden descargarse desde: <http://www.visual-paradigm> y, <http://argouml.tigris.org>.

Los requisitos mínimos de instalación para Visual Paradigm son 512MB de RAM (recomendado un 1GB) y 800MB de espacio en disco. ArgoUML requiere tan sólo 15MB de espacio en disco y la RAM necesaria para ejecutar Java.

Todas las herramientas necesarias para la asignatura se encuentran instaladas en los laboratorios del grado de informática.

Tutorías

Breve descripción

Tutorías individuales o colectivas: ayudan al alumno a aclarar dudas, están pueden ser colectivas, en el horario de clase habitual (horas destinadas a tutorías, ver plan de trabajo) o individuales, en el horario establecido de tutorías por la profesora o en otro horario acordado por la profesora/alumno previa cita del alumno por el campus virtual.

En las tutorías se resolverán ejercicios que refuercen los contenidos teórico-prácticos de la asignatura. El trabajo será evaluado por la profesora y los alumnos atendiendo siempre a la calidad general del trabajo y a las habilidades y actitudes expuestas.

Tutorías personales:

A todos los alumnos de la UCAM se les asigna un tutor personal del Cuerpo Especial de Tutores, cuando realizan su primera matrícula en la Universidad, de tal forma que el alumno recibe el acompañamiento de su tutor para toda su etapa universitaria según criterios y aspectos que se pueden consultar en: <http://www.ucam.edu/servicios/tutorias/preguntas-frecuentes/que-es-tutoria>.